EXHIBIT B

UNITED STATES DISTRICT COURT
FOR THE NORTHERN DISTRICT OF CALIFORNIA
SAN FRANCISCO DIVISION

ORACLE, INC.

Plaintiff,

v.

GOOGLE INC.,

Defendant.

Case No. CV 10-03561 WHA

HIGHLY CONFIDENTIAL

EXPERT REPORT OF DR. GREGORY K. LEONARD

CORRECTED (MARCH 10, 2016)

4. Conclusion Regarding the Contribution of the Alleged Infringement

173. Based on the foregoing, I conclude that the contribution of the alleged infringement has been small relative to the contributions of other factors that were substantially more important to Android's success, and thus to causing the revenues Mr. Malackowski claims are attributable to the infringement. This implies that the apportionment of the Android-related profits to the alleged infringement should be small as a matter of economics.

E. Quantitative Approaches to Apportionment

1. Bottom Up Approaches: Direct Measurement of the Android-Related Profits Attributable to the Alleged Infringement

174. A "bottom up" apportionment works by directly measuring the portion of the Android-related profits that are attributable to the alleged infringement. As discussed above, the contribution of the alleged infringement to Google's Android-related profits was that it generated cost-savings for Google by allowing Google to avoid taking certain costly actions such as licensing the allegedly infringing work under the OpenJDK and implementing the 37 API packages prior to the Android launch, paying for developers to be trained in another programming language, or paying for application development. I discuss several approaches to measuring these cost-savings. I also analyze the contribution of the alleged infringement to the Android-related profits under the assumption that Google took none of these costly actions. I find this measure to be small relative to the Android-related profits, but still larger than the cost-savings associated with the avoided actions described above. Thus, the appropriate apportionment should be based on the cost-savings and, in fact, the smallest of the cost-savings.

Thus, the cost-savings related to paying for app development is an alternative measure of the unjust enrichment (assuming infringement).

183. As discussed above, estimates of the costs of app development range from $25,000 to $100,000. As also discussed above, a relatively small number of apps account for the large majority of usage. Some of these apps would be written in C/C++ in any event (so that Google would not need to pay for development), and others would be written be developers (e.g., Facebook) who would not need any additional incentive to develop Android apps even in a different language. I estimate that, over the time period in question, there would be no more than 1000 "important" non-C/C++ apps.[277]

184. Assuming, quite conservatively, that Google would have had to pay for development of all these apps, the cost-savings to Google from the alleged infringement (and thus the unjust enrichment) would be between $23 million and $100 million, depending on the cost of app development.[278]

d. Incremental Android-Related Profit Attributable to the Alleged Infringement, Assuming No Costly Google Actions

185. As discussed above, the alleged infringement generated cost-savings for Google by allowing Google to avoid taking certain costly actions. If Google did not take such actions and did not allegedly infringe, Oracle's claim is that there would have been fewer Android apps

[277] Based on the comScore data, there are 428 unique apps that were among the monthly top 200 most used apps during the January to March 2013 and January to March 2015 periods combined. Using an analysis of how long apps tend to remain in the top 200, I estimate that no more than 1,900 apps would be among the monthly top 200 most used apps over the course of the Android's existence. I conservatively estimate that 65% of these apps would be Google apps, C++ apps, or apps by multi-homing developers. Thus, Google would have to cover the development cost for a maximum of 665 apps. I further conservatively round this up to 1000 apps.

[278] See Exhibit 3b.

and this would have impacted Android device sales and therefore the Android-related profits. In this section, I measure these effects.

186. I have applied the Kim (2013) empirical model of smartphone demand conservatively to estimate the decrease in Android handset sales that would have occurred in a counterfactual where there were fewer Android apps, as well as the percentage of this Android sales decrease that would have been captured by the iPhone. Google would earn ad revenue on these additional iPhone units.

187. In the model, the Android share satisfies (after application of the Berry (1994) inversion)[279]

$$\ln(s_A) - \ln(s_0) = \delta_A + \sigma \ln(s_{A|A,I})$$

where s_A is the Android share, s_0 is the share of the outside good, δ_A is the mean utility level for Android (based on the right-hand-side of equation (2.3) of Kim (2013)), and $s_{A|A,I}$ is Android's share of a nest consisting of Android and iPhone. iPhone share is defined analogously as

$$\ln(s_I) - \ln(s_0) = \delta_I + \sigma \ln(s_{I|A,I})$$

Blackberry share is defined as

$$\ln(s_B) - \ln(s_0) = \delta_B$$

and the share of the outside good is defined as $s_0 = 1 - s_A - s_I - s_B$.

188. Given data on shares and the value of σ[280], the three mean utility levels δ_A, δ_I, and δ_B can be solved for.

[279] S. Berry, Estimating Discrete-Choice Models of Product Differentiation, Econometrica (1994). In these equations, I omit notation related to time for clarity. The analysis is conducted on a monthly basis.

190. The counterfactual shares can be calculated by solving the following system of equations.

$$\ln(s'_A) - \ln(s'_0) = \delta'_A + \sigma \ln(s'_{A|A,I})$$

$$\ln(s'_I) - \ln(s'_0) = \delta_I + \sigma \ln(s'_{I|A,I})$$

$$\ln(s'_B) - \ln(s'_0) = \delta_B$$

$$s'_0 = 1 - s'_A - s'_I - s'_B$$

191. In the Kim (2013) model, the variables are defined as follows.

- The share of a smartphone OS in a given month is defined as the U.S. unit handset sales in that month divided by the U.S. population over the age of ten. I used U.S. unit handset sales data from ITG and U.S. population data from the U.S. Census.

- An app's share is defined as the number of downloads of the app in a given month divided by the handset sales of the OS in that month. I obtained app download data from AppAnnie. These data cover the period January 2012 to December 2015. Kim (2013) identified the set of available apps for Android and iPhone based on top paid and free app lists. I used the same approach.

192. For each Android app, I determined whether (1) it is a Google app, (2) it was written using the NDK, (3) it was multi-homed on iOS, (4) its developer also developed apps for iOS, or (5) its developer also developed NDK Android apps. Any app in one or more than one of these categories is assumed to be available on Android in the counterfactual. Apps in none of these categories are assumed to be unavailable in the counterfactual. This is conservative because apps not in one of these categories may well have been developed anyway. For example, many apps not in these categories have close counterparts on Microsoft Phone, which

to know the values of the other individual explanatory variables; their combined value is all that is needed and this can be determined through the Berry (1994) share inversion.